

SMTP TLS Settings

SMTP TLS Settings

Admin path: **System > SMTP TLS Settings** (`view_smtp_tls_settings.cfm`, `inc/get_smtp_tls_settings.cfm`, `inc/get_smtp_tls_policies.cfm`, `inc/edit_smtp_tls_settings.cfm`, `inc/smtp_tls_save_settings.cfm`, `inc/smtp_tls_add_domain.cfm`, `inc/smtp_tls_edit_domain.cfm`, `inc/smtp_tls_delete_domain.cfm`, `inc/generate_tls_policy.cfm`, `inc/generate_postfix_configuration.cfm`).

This page configures **Postfix TLS** end to end: the global inbound/outbound TLS mode (Disabled / Opportunistic / Mandatory), the certificate Postfix presents on `:25`/`:587`, and per-destination-domain TLS policy overrides for outbound delivery.

Pairs with [System Certificates](#), which owns the certificate **store**; this page is the **binding** of one of those certs to the Postfix `smtpd_tls_*` / `smtp_tls_*` directives. Pairs also with [Server Setup](#), which owns the SMTP banner hostname (`myhostname`) — the cert's Common Name or SAN must match that hostname for strict STARTTLS verifiers to accept the handshake.

TLS modes

+-----+		
Postfix smtpd_tls_security_level		
+ smtp_tls_security_level		
+-----+		
'' (Disabled)	'may' (Opportunistic)	'encrypt' (Mandatory)
v	v	v
no STARTTLS advertised (cleartext only)	STARTTLS offered; clear- text fallback if peer can't negotiate	STARTTLS required; peer must support it or delivery fails

Mode (<code>tlsmode</code> form value)	Postfix value	Use when
---	---------------	----------

Disabled ("")	(directive value cleared)	Cleartext-only environments (test, isolated networks); production Internet exposure not recommended
Opportunistic TLS (may) — Recommended	may	Standard public-Internet config. STARTTLS is advertised; peers that support it use it, peers that don't fall back to cleartext
Mandatory TLS (encrypt) — NOT recommended for Internet-facing servers	encrypt	Closed networks where every peer is known to support TLS. On the open Internet this drops mail from any sender that can't negotiate STARTTLS, which is a long tail of misconfigured small senders

The mode applies symmetrically to inbound (smtpd_*) and outbound (smtp_*). Both directive rows are written on save.

Selecting a certificate

The **SMTP TLS Certificate** field is a free-text autocomplete that searches system_certificates via the getcertificates.cfm ajax endpoint (the same endpoint used by Console Settings). Picking a row populates a hidden certificateno_1 field with the row ID plus four read-only display fields (Subject, Issuer, Serial, Type).

The certificate picker is **hidden when TLS mode is Disabled** (#tlscertificate div toggled by #tismode change handler). Switching back to Opportunistic or Mandatory slides it back into view.

The system-cert refusal

If an admin tries to save with the system-managed (bootstrap snakeoil) cert selected, the handler refuses with **error 3**:

“ You cannot select the system-self-signed Certificate for SMTP TLS.

This is intentional. A self-signed cert on :25 would defeat the purpose — strict STARTTLS verifiers on the receiving side reject the handshake, and Hermes would silently lose all outbound mail to those recipients. The refusal forces the admin to import a real cert (commercial CA, internal PKI, or Let's Encrypt) before flipping TLS on.

The error message text is dated — the comparison is against certificateno_1 = 1 in edit_smtp_tls_settings.cfm, which works on Docker fresh installs (where the bootstrap row is id =

1) but does **not** work on installs where the system cert was assigned a different ID (notably DEV's `ssl-cert-snakeoil` row at `id = 29`). The [System Certificates](#) runtime helper resolves this for the deletion guard; the SMTP-TLS save handler still uses the hardcoded `id = 1` check. Practical impact is small because in either case the admin should not be selecting the system row, but if you migrate from a legacy install with a non-`id=1` system row, the SMTP page won't refuse the snakeoil even though the System Certificates page will block its deletion.

How directive values are stored

This page is the canonical example of the dual-row `parameters` table pattern documented in [Server Setup § Configuration storage](#). Each Postfix directive has two rows:

Row	parameter	child	parent_name	Role
Name row	<code>smtpd_tls_security_level</code>	2	—	Directive name
Value row	<code>may / encrypt / ""</code>	1	<code>smtpd_tls_security_level</code>	Directive value

Save handler `edit_smtp_tls_settings.cfm` writes to the value row only:

```
UPDATE parameters
  SET parameter = '<tls_mode>'
 WHERE parent_name = 'smtpd_tls_security_level'
    AND child = '1'
    AND enabled = '1';

-- same for smtpd_tls_security_level (outbound)
-- and smtpd_tls_cert_file, smtpd_tls_key_file, smtpd_tls_CAfile
-- (paths resolved from system_certificates.file_name + type)
```

The selected cert's on-disk paths are derived from `system_certificates.type` + `file_name`:

type	smtpd_tls_cert_file	smtpd_tls_key_file	smtpd_tls_CAfile
Imported	<code>/opt/hermes/ssl/<file_name>_hermes.pem</code>	<code>/opt/hermes/ssl/<file_name>_hermes.key</code>	<code>/opt/hermes/ssl/<file_name>_hermes.chain.pem</code>
Acme	<code>/etc/letsencrypt/live/<file_name>/cert.pem</code>	<code>/etc/letsencrypt/live/<file_name>/privkey.pem</code>	<code>/etc/letsencrypt/live/<file_name>/chain.pem</code>

The same path-derivation logic is implemented globally in `inc/get_active_cert_paths.cfm` for the console binding; the SMTP save handler open-codes it here (technical debt — the path arithmetic should be moved to the helper so there's only one place that knows the layout).

The new directive values land in the `parameters` table, then `generate_postfix_configuration.cfm` regenerates `main.cf` from the live rows and runs `postfix reload`. Mode changes therefore take effect on the next SMTP connection without dropping in-flight sessions (`postfix reload` is a `SIGHUP`, not a restart).

What this page does NOT configure

Hermes' TLS surface is opinionated by design. The page deliberately omits several knobs that Postfix exposes:

Concern	Status
Cipher suite (<code>smtpd_tls_ciphers</code> , <code>smtpd_tls_mandatory_ciphers</code>)	Hardcoded in <code>main.cf</code> baseline; no UI
Protocol versions (<code>smtpd_tls_protocols</code> , <code>smtpd_tls_mandatory_protocols</code>)	Hardcoded in <code>main.cf</code> baseline; no UI
DH parameters (<code>smtpd_tls_dh1024_param_file</code>)	Same ECDHE-only decision as Console Settings — DH is not offered
TLS session cache	Hardcoded defaults
EECDH curve	Hardcoded defaults
Per-mailbox-domain certs (autoconfig/autodiscover)	Lives on SAN Management ; this page binds the single cert Postfix presents on the public SMTP banner
Dovecot IMAP/POP cert	Email Server > Settings (separate <code>mail.certificate</code> binding)
Console (nginx) cert	Console Settings

The cipher / protocol decisions are baked into the Postfix baseline config because they have global security implications and changing them needs more than a dropdown — there's no curated "modern / intermediate / legacy" preset UI yet, and the right defaults for an SEG track [Mozilla's modern profile](#) which doesn't churn often enough to warrant operator-tunable UI.

TLS Policy Domains — per-destination outbound overrides

Below the global card is the **TLS Policy Domains** table. Each row forces a stricter-than-global TLS policy for outbound mail to a specific recipient domain.

Field	Meaning
Domain	Recipient domain (<code>example.com</code>) or domain-and-subdomains pattern (<code>.example.com</code> — leading dot matches all subdomains)
Encryption Mode	Currently always Mandatory (<code>encrypt</code>) for manually-added rows. Per-row mode tunables are tracked but not exposed.
Note	Free-text description shown in the row

Adding a row generates `/etc/postfix/tls_policy` (via `generate_tls_policy.cfm`), runs `postmap` to compile it into a hash map, and reloads Postfix:

```
docker exec hermes_postfix_dkim /usr/sbin/postmap /etc/postfix/tls_policy
```

The Postfix daemon then consults the map for every outbound SMTP connection — entries matching the destination domain override `smtp_tls_security_level` for that specific destination.

“ **Operational consequence.** Adding a `encrypt` policy for a recipient domain whose MX **doesn't actually support STARTTLS** silently breaks outbound mail to that domain. Postfix will defer + bounce. Verify the recipient MX advertises STARTTLS before adding a Mandatory entry. The warning callout on the page itself spells this out.

Auto-added rows (managed by Domains)

When a domain on Email Server > Domains or Email Relay > Domains is configured to require SASL authentication, Hermes auto-inserts a TLS policy row to enforce encryption for that destination. These rows are marked by `description = 'Auto-added: domain requires authentication'` and rendered with a special **Managed by Domains** badge:

- The row's **checkbox** is suppressed (cannot be bulk-deleted from here)
- The row's **Edit** button is suppressed (must be edited on the managing page)
- The Note column links to **view_domains.cfm** so the admin lands on the right page

This is the same pattern used elsewhere in Hermes for system-owned rows that would otherwise look user-editable — surface that the row is managed somewhere else and link to the managing page.

Save flows

Save SMTP TLS Settings (`save_settings`)

1. Validate `form.tlsmode` in ("", "may", "encrypt")
2. UPDATE parameters value rows for `smtpd_tls_security_level` + `smtp_tls_security_level`
3. If `tlsmode` is not "" :
 - a. Validate `certificateno_1` exists in `system_certificates`
 - b. Refuse if `certificateno_1` = 1 (legacy bootstrap-id check)
 - c. UPDATE parameters2 `smtp.certificate`
 - d. Derive cert/key/CA paths from `type` + `file_name`
 - e. UPDATE parameters value rows for `smtpd_tls_cert_file` / `smtpd_tls_key_file` / `smtpd_tls_CAfile`
4. `generate_postfix_configuration.cfm` (regenerate `main.cf` + postfix reload)
5. `session.m` = 35 ("settings saved successfully. Postfix reloaded.")
6. cflocation back to `view_smtp_tls_settings.cfm`

Add / Edit / Delete TLS Policy Domain (`add_domain` / `edit_domain` / `delete_domain`)

1. Validate domain (email-trick: `IsValid("email", "bob@<domain>")`)
 - Leading "." accepted; validator prepends "subdomain"
2. INSERT / UPDATE / DELETE in `tls_policies`
3. `generate_tls_policy.cfm` (rewrite `/etc/postfix/tls_policy` + `postmap`)
4. `generate_postfix_configuration.cfm` (postfix reload)
5. `session.m` = 37 / 39 / 34 (per action)
6. cflocation back to `view_smtp_tls_settings.cfm`

Both save flows end in a `postfix reload`, which is a SIGHUP — no in-flight SMTP connections are dropped, and queued mail continues delivering normally.

Failure semantics

What breaks	What happens
Mode = Opportunistic/Mandatory + Certificate empty	m = 1, "SMTP TLS Certificate cannot be blank when TLS Mode is set to Opportunistic or Mandatory"
Certificate ID does not exist in <code>system_certificates</code>	m = 2, "The SMTP TLS Certificate you entered is not valid"
Certificate ID is 1 (legacy bootstrap check)	m = 3, "You cannot select the system-self-signed Certificate for SMTP TLS"
Domain validation fails on add/edit	m = 4
Duplicate domain on add	m = 5
Duplicate domain on edit	m = 6
Missing required form field	m = 20
<code>generate_tls_policy.cfm</code> fails (cp / mv / postmap)	DB is ahead of the live <code>tls_policy.db</code> . Next save re-renders cleanly. The previous live map is preserved as <code>/etc/postfix/tls_policy.HERMES.BACKUP</code> .
<code>postfix reload</code> fails inside the container	DB and on-disk config in sync; running daemon stale. Recovery: <code>docker exec hermes_postfix_dkim postfix reload</code> manually.

Files and containers touched

Path	Owner	Role
<code>config/hermes/var/www/html/admin/2/view_smtp_tls_settings.cfm</code>	<code>hermes_commandbox</code>	Page
<code>config/hermes/var/www/html/admin/2/inc/edit_smtp_tls_settings.cfm</code>	<code>hermes_commandbox</code>	Save handler (mode + cert binding)
<code>config/hermes/var/www/html/admin/2/inc/smtp_tls_save_settings.cfm</code>	<code>hermes_commandbox</code>	Action handler wrapper around <code>edit_smtp_tls_settings.cfm</code>
<code>config/hermes/var/www/html/admin/2/inc/smtp_tls_add_domain.cfm</code>	<code>hermes_commandbox</code>	TLS Policy add
<code>config/hermes/var/www/html/admin/2/inc/smtp_tls_edit_domain.cfm</code>	<code>hermes_commandbox</code>	TLS Policy edit
<code>config/hermes/var/www/html/admin/2/inc/smtp_tls_delete_domain.cfm</code>	<code>hermes_commandbox</code>	TLS Policy delete
<code>config/hermes/var/www/html/admin/2/inc/generate_tls_policy.cfm</code>	<code>hermes_commandbox</code>	Render <code>/etc/postfix/tls_policy</code> + <code>postmap</code>
<code>config/hermes/var/www/html/admin/2/inc/generate_postfix_configuration.cfm</code>	<code>hermes_commandbox</code>	<code>main.cf</code> regen + <code>postfix reload</code>
<code>/etc/postfix/main.cf</code>	<code>hermes_postfix_dkim</code> (mounted)	Live Postfix config — regen target
<code>/etc/postfix/tls_policy</code> + <code>tls_policy.db</code>	<code>hermes_postfix_dkim</code> (mounted)	Live TLS-policy map (text + postmap-compiled)

Path	Owner	Role
<code>/etc/postfix/tls_policy.HERMES.BACKUP</code>	<code>hermes_postfix_dkim</code> (mounted)	Write-time backup of the previous live map
<code>parameters</code> rows for <code>smtpd_tls_*</code> and <code>smtp_tls_*</code>	<code>hermes_db_server</code> (<code>hermes</code> DB)	Directive values
<code>parameters2.smtp.certificate</code>	<code>hermes_db_server</code> (<code>hermes</code> DB)	Active SMTP cert binding (FK into <code>system_certificates.id</code>)
<code>tls_policies</code> table	<code>hermes_db_server</code> (<code>hermes</code> DB)	Per-destination overrides

Every shell-out uses `docker exec hermes_postfix_dkim ...` per the standard Hermes Docker pattern. `postmap` is the one operation that absolutely **must** run inside the container — the `tls_policy.db` hash format is libdb-version-sensitive, and running it on the host produces a file Postfix inside the container can't read.

Related

- [System Certificates](#) — the certificate store this page selects from; system-managed certs cannot be bound for SMTP
- [Server Setup](#) — Mail Server Hostname (`myhostname`); the SMTP cert's Subject CN or SAN should match this for strict STARTTLS verifiers
- [Console Settings](#) — the console-side analogue of this page (binds a System Certificate to nginx)
- [Authentication Settings](#) — Authelia / SASL; per-domain SASL requirements auto-insert `tls_policies` rows here
- [LDAP RemoteAuth](#) — upstream LDAP TLS settings; separate CA store at `/opt/hermes/certs/remotearch/`, not part of System Certificates
- [SAN Management](#) — per-mailbox-domain certs for autodiscover/autoconfig; orthogonal to the single SMTP cert this page binds
- [Intrusion Prevention](#) — Fail2ban; not TLS-related but relevant for hardening the SMTP service this page configures
- [Admin Console Firewall](#) — IP allowlist for the console (not SMTP); SMTP is open to the Internet for inbound mail

Revision #48

Created 2026-05-31 12:52:02 UTC by Dino Edwards

Updated 2026-06-20 13:33:01 UTC by Dino Edwards